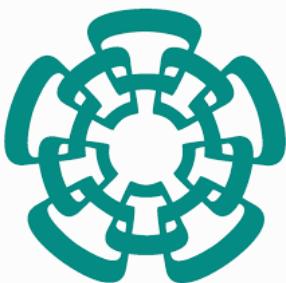


# **Centro de Investigación y Estudios Avanzados del I.P.N.**



## **Reconocimiento de 8 herramientas**

Materia: Análisis de Imágenes Digitales

Catedrático: Dr. Wilfrido Gómez-Flores

Alumno: Luis Alberto Ballado Aradias

29 de agosto de 2023

# Índice

|          |                                       |           |
|----------|---------------------------------------|-----------|
| <b>1</b> | <b>Introducción</b>                   | <b>2</b>  |
| <b>2</b> | <b>Adquisición de la imagen</b>       | <b>3</b>  |
| <b>3</b> | <b>Preprocesamiento</b>               | <b>5</b>  |
| <b>4</b> | <b>Segmentación</b>                   | <b>6</b>  |
| 4.1      | Operaciones morfológicas . . . . .    | 9         |
| 4.2      | Erosión y dilatación . . . . .        | 10        |
| <b>5</b> | <b>Extracción de características</b>  | <b>10</b> |
| 5.0.1    | Geométricas básicas . . . . .         | 11        |
| 5.0.2    | Rasgos basados en esqueleto . . . . . | 12        |
| 5.0.3    | Momentos invariantes Hue . . . . .    | 12        |
| 5.0.4    | Cerco convexo . . . . .               | 12        |
| <b>6</b> | <b>Clasificación</b>                  | <b>13</b> |
| <b>7</b> | <b>Resultados</b>                     | <b>14</b> |
| <b>8</b> | <b>Conclusiones</b>                   | <b>15</b> |

## 1. Introducción

El procesamiento de imágenes es un área multidisciplinaria que tiene como objetivo la manipulación y análisis de la información de una imagen en su parte más reducida conocida como pixel (picture element). Emergiendo una gran cantidad de técnicas y metodologías para el procesamiento y manipulación de imágenes.

Con el aumento de imágenes digitales, el procesamiento se ha vuelto necesario para aplicaciones como en robótica, que la segmentación de imágenes puede facilitar la navegación para un robot aéreo [6], en el diagnóstico de cáncer ayuda a los médicos en dar diagnósticos más acertados [4]. Asimismo, en la parte del entrenamiento con implementación de realidad aumentada [1] .

La detección y reconocimiento que son parte del área de visión por computadora, se han vuelto cada vez más utilizadas en aplicaciones como traductores de idiomas, análisis médicos, reconocimiento de rostros faciales.

La detección y categorización de objetos con imágenes se descompone de los siguientes procesos:

1. **Adquisición de la imagen.-** La toma de fotografía se realiza mediante una cámara que cuente con un sensor que puede ser de tipo CMOS, el sensor convierte la captura de la escena en señales digitales que son guardadas como una imagen.
2. **Preprocesamiento.-** Mediante las técnicas existentes se busca el mejoramiento del contraste, realzar características reduciendo el ruido, para hacer de la imagen lo más limpia posible para el paso de segmentación.
3. **Segmentación.-** Es la obtención de las regiones de interés que ayuda a la extracción de características. Ésta imagen binaria debe tomar la silueta de la figura lo más aproximada posible.
4. **Extracción de características.-** Éste proceso busca describir en cierta forma la imagen, describiendo valores de forma geométrica, colores, texturas y valores invariantes que se mantienen a pesar de rotaciones o escalamientos.
5. **Clasificación.-** Asigna una clase a cada región de interés dentro de la imagen basados en los rasgos extraídos en el paso anterior.

El proyecto busca el reconocimiento de ocho herramientas, aplicando las técnicas vistas en clase con el uso de MATLAB.

La figura 1 muestra las etapas a desarrollar para el reconocimiento de las ocho herramientas.

Una imagen  $f(x, y)$  se puede modelar como el producto de dos componentes  $f(x, y) = i(x, y), r(x, y)$ , donde  $0 < i(x, y) < \infty$  es la componente de iluminación y  $0 < r(x, y) < 1$  es la componente de reflectancia.

La dificultad en el procesamiento de imágenes digitales está representada por la luz ambiental, ya que el sensor percibe la iluminación al grabar la imagen.

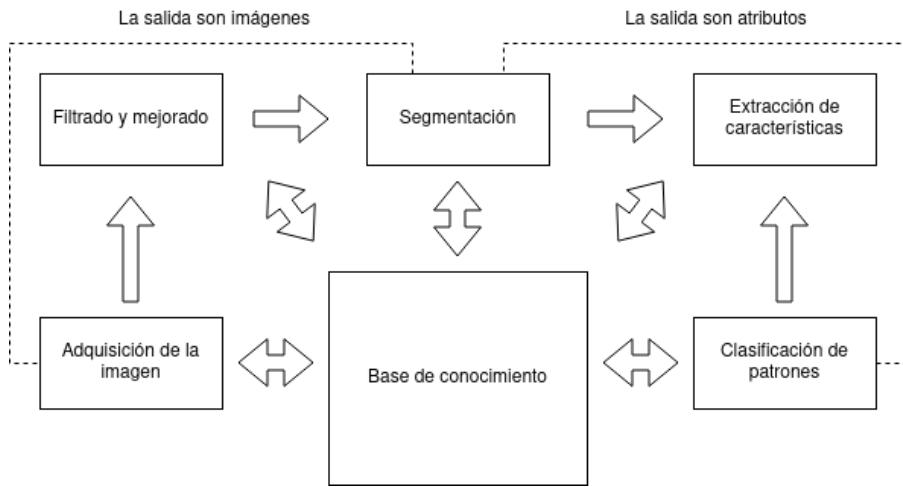


Figura 1: Etapas básicas para el análisis de imágenes.

## 2. Adquisición de la imagen

La adquisición de las imágenes se realizó con las siguientes herramientas:

- Cámara web Logitech C920s HD Pro 1080p webcam
- Cubo para la toma de fotografías marca Puluz, ver figura 3
- Control de iluminación dimmer, ver figura 2



Figura 2: Dimmer para el ajuste de la iluminación.

La cámara web cuenta con un sensor de tipo CMOS de 2-mega pixel [2], obteniendo imágenes limpias y muy bien contrastadas.

El objetivo del proyecto, es el reconocer automáticamente ocho herramientas que son:

- Martillo
- Desarmador
- Cinta de medir
- Llave perica
- Tijera
- Pinza de punta
- Pinza eléctricas
- Pinza de presión

La toma de fotografías se realiza en un ambiente controlado como se muestra en la figura 3, ajustando la iluminación con un dimmer mostrado en la figura 2.



Figura 3: Cubo para la adquisición de imágenes.

Se usaron transformaciones geométricas para modificar la relación espacial entre píxeles en una imagen, usando propiedades que no deformen al objeto de interés como la rotación. Se cuidó en rellenar los valores del fondo con unos (1), para evitar los valores oscuros por los ceros que se puedan presentar por la rotación.

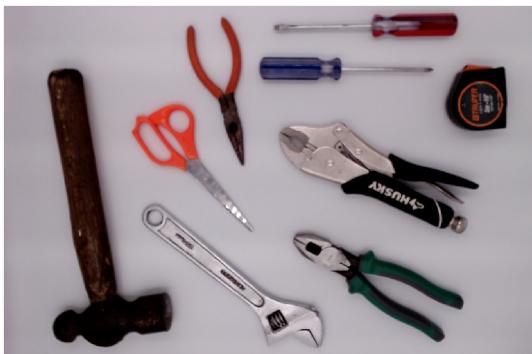
Logrando crear imágenes artificiales, de manera que sirvan para el aumento del conjunto de imágenes.

### 3. Preprocesamiento

Al contar con imágenes de buena calidad, solo se mejoró el contraste para reducir las sombras que las herramientas pueden formar y lograr obtener el contorno que representa más la imagen a reconocer.

El mejoramiento, es la manipulación de una imagen, de tal forma que el resultado sea más útil que la imagen original para una aplicación en particular.

Una transformación de intensidad nos ayuda a modificar el contraste. Dentro del proyecto se aplicó una transformación de intensidad con un escalar de 1.5. Para algunas imágenes más oscuras el valor de 2 resulta mejor para la detección de bordes más cercanos al objeto. El problema de obtener la mejor intensidad de brillo se puede optimizar en base a los desempeños obtenidos mediante alguna estrategia de optimización.



(a) Imagen original



(b) Imagen modificada

Figura 4: Mejoramiento de contraste.

De esta forma pudimos darle un alto contraste a la imagen como se puede apreciar en la figura 4.

A partir de la imagen original pasada a double<sup>1</sup>, esta imagen se multiplica por un escalar de brillo para tener la imagen con el brillo alto.

```
1 %% pasar a double
2 original_gris = double(I);
3 %% multiplicar por el factor de ruido
4 img_ajustada = original_gris * factor_brillo;
```

Código 1: Factor de brillo

A pesar de que los objetos en su mayoría poseen partes cromadas, se logran contrastar con el fondo original blanco. Logrando apreciar mejor los objetos. Éste paso es de gran ayuda, siendo el factor de brillo un atributo dentro de la función de segmentación.

<sup>1</sup>se hace un cambio de estructura a double

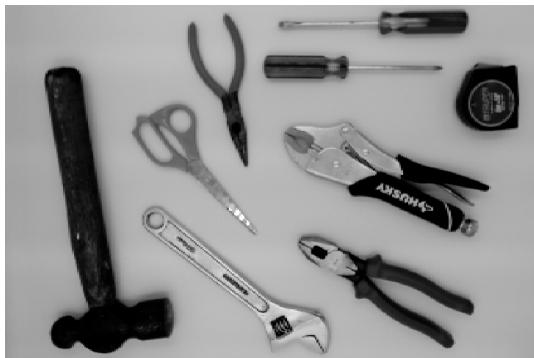
## 4. Segmentación

Se hacen pruebas con diversas técnicas de segmentación, optando por un resultado sencillo con uso de operaciones morfológicas en base a una imagen de detección de bordes con Canny Edges.

Previo a la estrategia de segmentación propuesta, se exploraron otras técnicas que se listan a continuación.

- Método de umbralado global **Método de Otsu**

Éste método sería el indicado, pero las partes color cromo en los objetos afecta en el umbralado de la imagen, haciendo perder rasgos representativos de los objetos como las puntas en la tijera, las partes metálicas del desarmador o de las pinzas. La figura 5 ilustra el resultado después de aplicar éste método (Otsu).



(a) Imagen original escala de grises



(b) Imagen umbralada con el método de otsu

Figura 5: Umbralado con método de otsu.

- Método de segmentación con **K-Means** (K-Means clustering)

El algoritmo k-means es una técnica de agrupamiento que divide un conjunto de datos en k grupos o clusters. Los datos se agrupan de modo que los puntos de un grupo sean más similares que los puntos de otros grupos.

A partir de una simple idea de crear dos grupos, el fondo y los objetos. Éste método sirve igual de bien que el Método de Otsu, pero en nuestro caso, el problema con el color cromo de los objetos hace que los valores cromo formen parte del conjunto del fondo. La figura 6 ilustra el resultado después de aplicar éste método (K-Means clustering).

- Visualización de la **Entropía**

Con una segmentación mediante el método de entropía, se puede obtener la información de la presencia de los objetos. Lamentablemente al estar con mucha información no fué utilizada en el proyecto. Pero replanteandolo nuevamente, se puede hacer algún filtro para quedarse solo con los píxeles blancos con alguna técnica como lógica difusa. La figura 13 ilustra el resultado después de aplicar éste método (Entropía).



(a) K-Means con dos clases



(b) K-Means con tres clases

Figura 6: Umbralado con K-Means.



(a) Imagen original escala de grises



(b) Imagen después del método de entropía

Figura 7: Umbralado con método de entropía.

- Detección de bordes **Canny Edges**

Al contar con una representación limpia de los objetos de interés, la detección de los objetos a partir de sus bordes es una buena opción para lidar con las piezas cromadas. La figura 8 ilustra el resultado después de aplicar éste método.

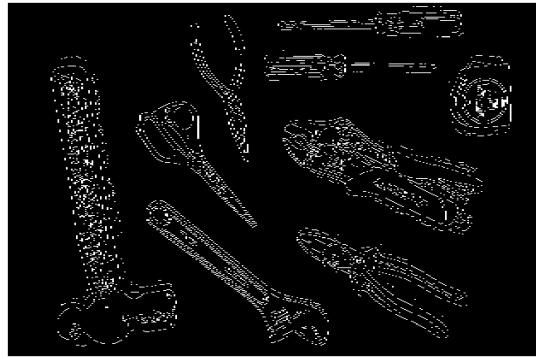
Probablemente sea el algoritmo para detección de bordes más usado para reconocer bordes en visión por computadora.

Usa las mejores propiedades del operador del gradiente y el operador laplaciano. Básicamente los pasos que sigue son los siguientes:

- Suaviza la imagen con un filtro gausiano 2D
- Calcula el gradiente de la imagen usando un operador de sobel de 3x3, dando resultados de las derivadas para cada punto para el eje x y el eje y
- A partir de esos valores se puede calcular la magnitud para cada píxel, así como su orientación
- Se calcula el Laplaciano a lo largo de la dirección del gradiente
- Se usan los cruces por cero en el Laplaciano para encontrar la ubicación del borde



(a) Imagen original escala de grises



(b) Detección de bordes con canny edges

Figura 8: Detección de bordes con canny edges.

Este método utiliza dos umbrales para detectar bordes fuertes y débiles e incluye bordes débiles en la salida si están asociados con bordes fuertes. Debido a que el método de Canny utiliza dos umbrales diferentes, es menos propenso a errores debido al ruido que otros métodos y es más probable que detecte bordes realmente débiles [5].

- Sigma valor escalar que especifica la desviación estándar del filtro gaussiano. El valor predeterminado es  $\text{sqrt}(2)$ . automáticamente en función de sigma [5].
- Umbral de sensibilidad es especificado como escalar numérico, la función en matlab ignora todos los bordes cuya intensidad no es mayor que threshold indicado, al no indicar un valor la función escoge los valores automáticamente [5].

Al considerar la última técnica con la obtención de los bordes, debemos cerrar la imagen con elementos estructurantes, donde podemos encontrar la línea, disco, diamante, cuadrado, rectángulo.

La segmentación se obtuvo a partir de una técnica hibrida en base a detectar los bordes con la técnica de Canny Edges. A continuación se mencionan los pasos que se siguieron.

1. Se cierra la imagen con un elemento estructurante de tipo square de 5 píxeles
2. Se rellenan los huecos
3. Se abre la imagen con uso de `bwareaopen` para quedarnos con elementos mayores a 2000 píxeles
4. Se remueven elementos espurios a partir de una erosión con un elemento estructurante de disco de 4 píxeles
5. Se abre la imagen con uso de `bwareaopen` para quedarnos con elementos mayores a 2000 píxeles
6. Se dilata la imagen con un elemento estructurante de disco de 2 píxeles
7. Se abre nuevamente la imagen con uso de `bwareaopen` para quedarnos con elementos mayores a 2000 píxeles

de esta forma artesanal, garantizamos en tener cubierta la forma de nuestro objetos de interés y pudimos dar solución a los objetos cromados.

#### 4.1. Operaciones morfológicas

La erosión y dilatación, operadores que fueron aplicados en el proyecto.

- Apertura: una erosión seguida de una dilatación  $f \circ b = (f \ominus b) \oplus b$
- Cerradura: una dilatación, seguida de una erosión  $f \bullet b = (f \oplus b) \ominus b$

El borde de un conjunto A que contiene los píxeles del primer plano se obtiene como:

$$B(A) = A - (A \ominus b)$$

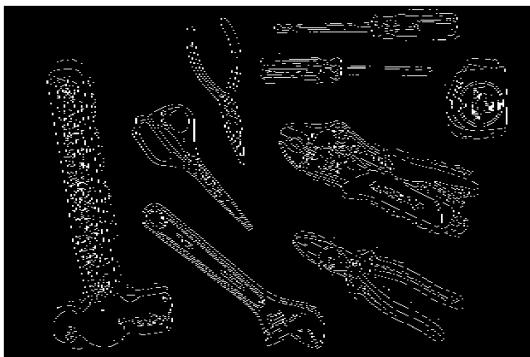
donde A es la figura binaria y b el elemento estructurante.

Un orificio es una región del fondo rodeado por un borde conexo de píxeles del primer plano. Una cerradura morfológica en la imagen binaria utilizando un elemento estructurante de tipo cuadrado de 5 píxeles.

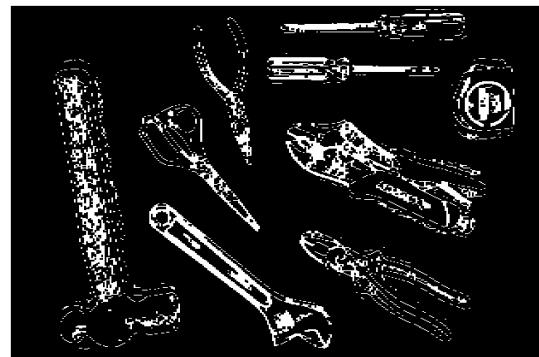
Rellenar huecos de la imagen binaria. Un hueco es un conjunto de píxeles de fondo que no se puede alcanzar llenando el fondo desde el borde de la imagen.

```
1 %CERRAR CON ELEMENTO ESTRUCTURANTE SQUARE 5
2 cerrar_imagen = imclose(imgBordes,strel('square',5));
3 %LENAR HUECOS
4 rellenar_imagen = imfill(cerrar_imagen,'holes');
```

Código 2: Cerradura seguido de llenado de huecos.



(a) Imagen después de Canny Edge y de efectuar una cerradura



(b) Imagen después de llenar huecos

Figura 9: Antes y después de la cerradura y llenado de huecos.

Dentro del proyecto, una de las operaciones más recurridas es la apertura del área para eliminar los elementos menores a 2,000 píxeles de la imagen binaria.

```
1 %QUITAR ELEMENTOS MENORES A 2000px
2 imagen_segmentada = bwareaopen(rellenar_imagen,2000);
```

Código 3: Apertura de 2000px

## 4.2. Erosión y dilatación

En unas versiones tempranas de la estrategia seguida, se contaban con uniones de los objetos de interés. Esto se pudo resolver erosionando la imagen binaria para después eliminar algunos objetos espurios dentro de la imagen.

```
1 %EROSIONAR PARA LIMPIAR MAS LA IMAGEN
2 erode_image = imerode(imagen_segmentada,strel('disk',4));
```

Código 4: Erosión con elemento estructurante disco 4px

Se vuelve a efectuar una apertura del área, buscando eliminar elementos menores a 2,000 píxeles.

```
1 %QUITAR ELEMENTOS MENORES A 2000px
2 mascara = bwareaopen(erode_image,2000);
```

Código 5: Apertura de 2000px

Se hace uso de una dilatación para dar forma al objeto de interés que deseamos reconocer.

```
1 %DILATAR PARA PRESERVAR LA FORMA DEL OBJETO
2 PP = imdilate(mascara,strel('disk',2));
3 P = bwareaopen(PP,2000);
```

Código 6: Dilatación con elemento estructurante disco 2px



(a) Imagen antes de la erosión con elementos espurios



(b) Imagen después de la apertura de 2,000px

Figura 10: Antes y después del último paso en la segmentación.

## 5. Extracción de características

El reconocimiento automático de objetos requiere calcular rasgos que describan propiedades físicas de los objetos.

Un rasgo (atributo o característica), es un valor numérico que cuantifica alguna propiedad de forma, textura, color, geometría, etc.

- Un rasgo debe ser discriminante, invariante, incorrelado y rápido de calcular.
- Los rasgos de forma generalmente se calculan a partir de la segmentación del objeto y se divide entre todos los pixeles de las regiones de interés.

La extracción de características que se utilizaron en el proyecto son las siguientes:

1. Rasgos geométricos básicos
2. Momentos de Hue
3. Cerclo convexo
4. Esqueletización

Se creó un script llamado **script\_rasgos.m** el cual itera dentro de la carpeta llamada `data_aumentation`, la cual consiste en las imágenes del dataset. Al considerarse el número de las carpetas, y si se desea crear una nueva clase, solo bastará con crear una nueva carpeta con el nombre de la nueva clase y poner las imágenes a extraer rasgos dentro de ella. Al final, el script calcula los rasgos a partir de la opción que haya elegido el usuario y guardándola en una extensión `.mat`.

#### 5.0.1. Geométricas básicas

El área es el número total de pixeles que cubre la región del objeto. El perímetro es la longitud del contorno del objeto. El resultado depende del tipo de conectividad que pueda tener. (4 ó 8 vecindad).

Para el cálculo de rasgos geométricos básicos, se hace uso del área y perímetro de la figura, el cual se obtiene mediante la imagen segmentada iterando entre todas las etiquetas.

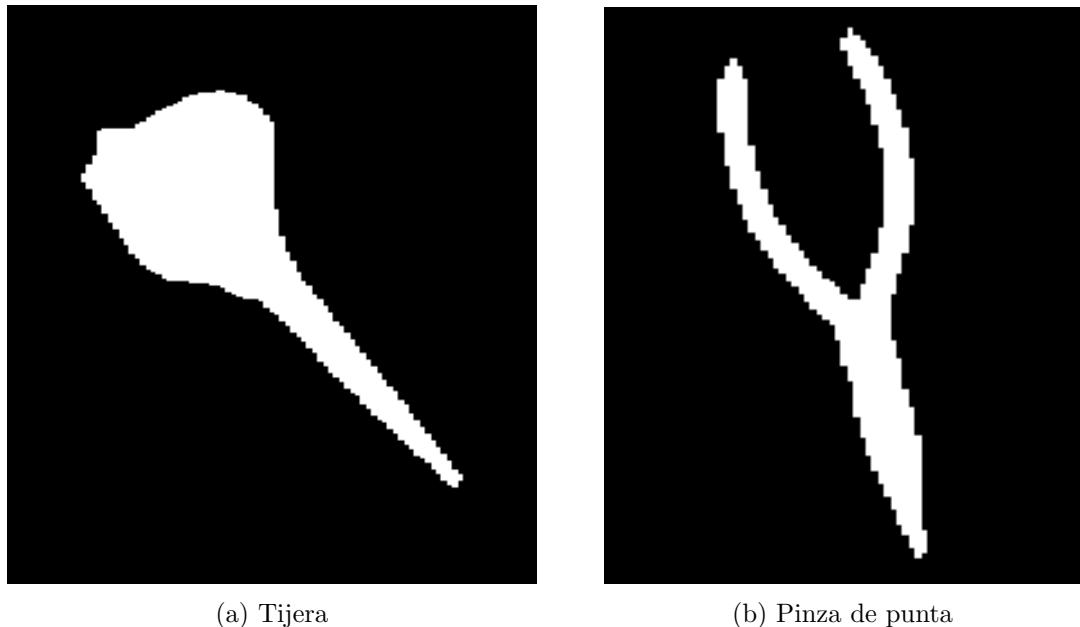
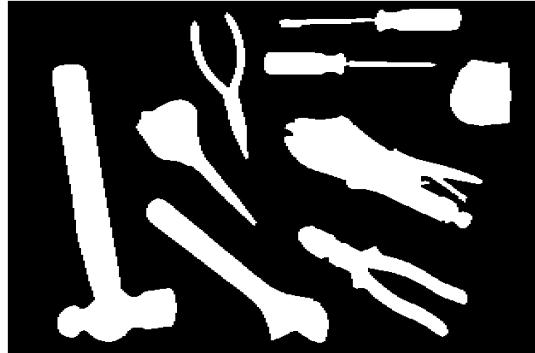


Figura 11: Imágenes segmentadas para obtención de rasgos geométricos básicos.

- Redondez
- Circularidad
- Compacidad
- Factor de forma

### 5.0.2. Rasgos basados en esqueleto

Otra generación de rasgos usada en el proyecto, son los rasgos a partir de esqueletizar la imagen binaria.



(a) Imagen umbralizada



(b) Imagen esqueletizada

Figura 12: Rasgos basados en esqueleto.

Se calculan los siguientes rasgos en base a la imagen esqueletizada.

- Puntos finales
- Ramas
- Número de pixles
- Área
- Perímetro

### 5.0.3. Momentos invariantes Hue

Un momento es la proyección de una función sobre una base polinómica utilizada para medir la distribución de masa de un objeto.

En el proyecto, se apoyo en revisar cómo funcionaba el código. Basandome en el siguiente código [7]

### 5.0.4. Cercos convexos

Las características basadas en bordes convexos comparan un objeto con sus bordes convexos para identificar regiones cóncavas o sobrantes. Cuanto más irregular es la forma del objeto, más espacio queda. Los rasgos que se obtienen son la solidez y la convexidad.

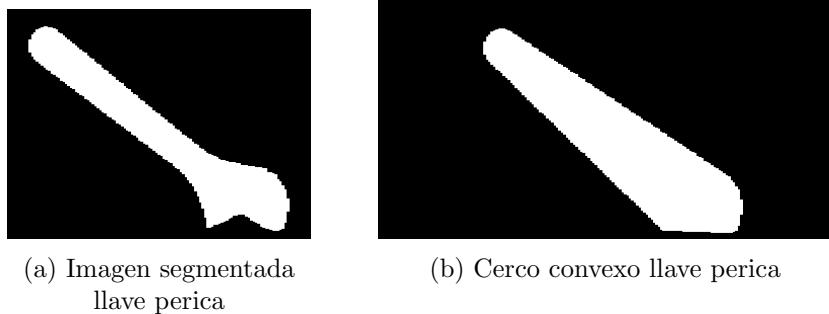


Figura 13: Ejemplo llave perica cerco convexo.

## 6. Clasificación

Recordando las etapas del reconocimiento de objetos en imágenes.

1. Sensado.- Capturar la imagen.
2. Preprocesamiento.- Mejorar la calidad de la imagen y segmentar los objetos de interés.
3. Extracción de características.- Describir los objetos con rasgos cuantitativos discriminantes e invariantes.
4. Clasificación.- Asignar una etiqueta de clase a cada objeto.

Obtener un vector de patrones.

En este proyecto se usa el método de datos de entrenamiento que son vectores de patrones asociados a una etiqueta de clase  $(x,y)$  con  $y \in \Omega = w_1, \dots, w_c$

De modo que el reconocimiento de un objeto se realiza mediante la comparación entre el objeto de entrada y las muestras que se hayan obtenido en la etapa de extracción de características.

Al pasar por su segmentación y analizar los rasgos obtenidos, se asigna al grupo de mayor parecido. Entonces, el objeto cercano indica que son parecidos y la similitud se reducirá a medida que no tenga una similitud.

Diferentes distancias se pueden implementar en el algoritmo de los k-vecinos más cercanos, siendo la distancia euclíadiana la que se utilizó en el desarrollo del proyecto. El parámetro k de vecinos, es el número de muestras de entrenamiento cercanas con las que se comparan, eligiendo k=4 para nuestro proyecto.

El algoritmo KNN realiza los siguientes pasos:

1. Medir la distancia Euclíadiana entre el objeto de interés y las muestras obtenidas la extracción de rasgos.
2. Identificar los patrones k más cercanos y obtener la etiqueta de la clase a la que pertenece.
3. Asignación de la clase al objeto de interés.

## 7. Resultados

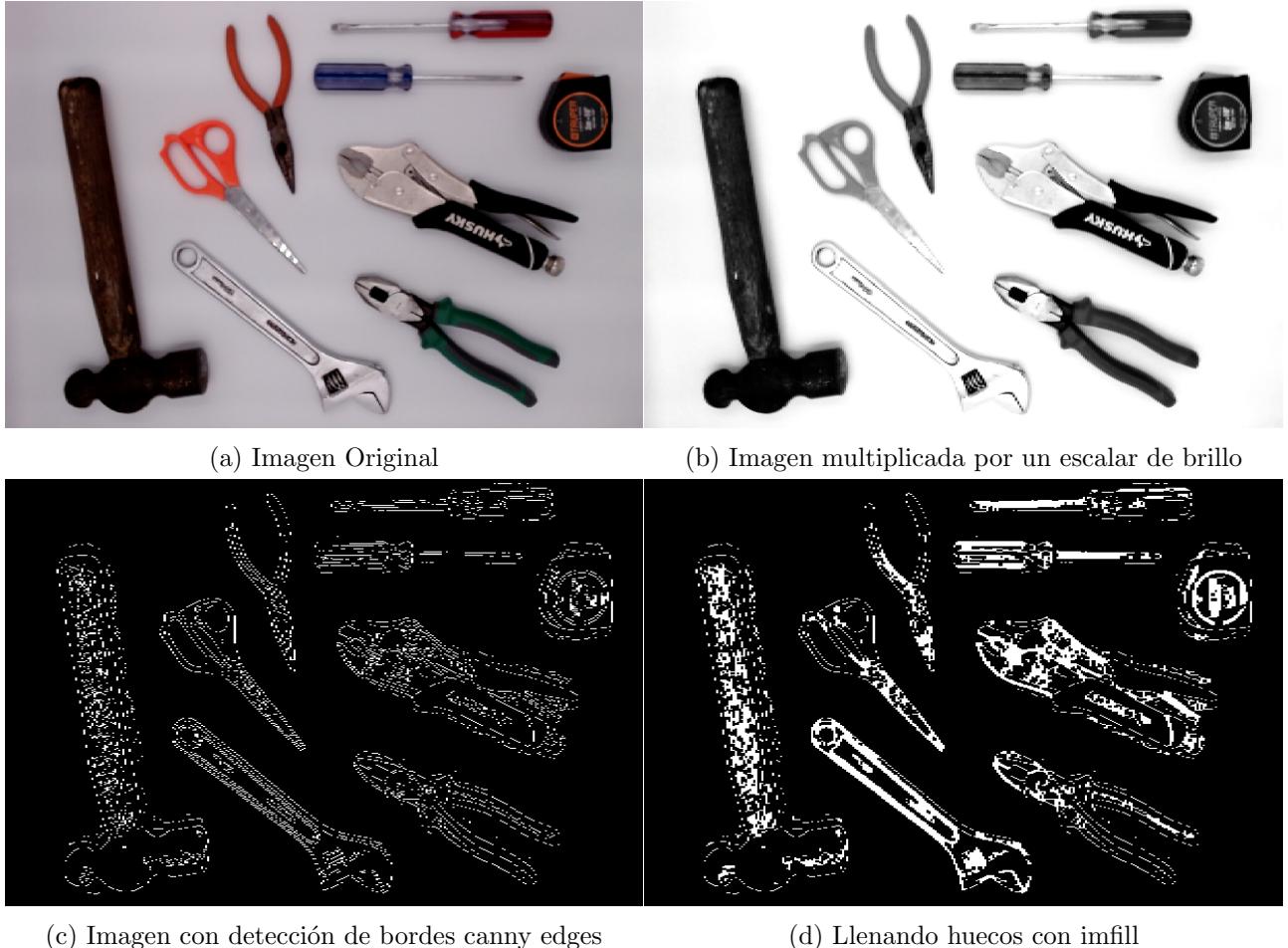


Figura 14: Proceso de segmentado primera parte.

Dentro de las imágenes, la similitud de esqueletos entre la cinta, la llave perica, tijera, pinzas de punta y electrica que tienen formas similares.

La mayoría de desarmadores no tiene ramas y tiene dos puntos finales, lo que lo confunde con una cinta al solo considerar las ramas y los puntos terminales se tiene una exactitud del 20 %, mientras al agregar el número de píxeles crece al 78 %, agregando más información como el área y perímetro la probabilidad de acierto crece aún más, presentándose valores arriba del 90 %.

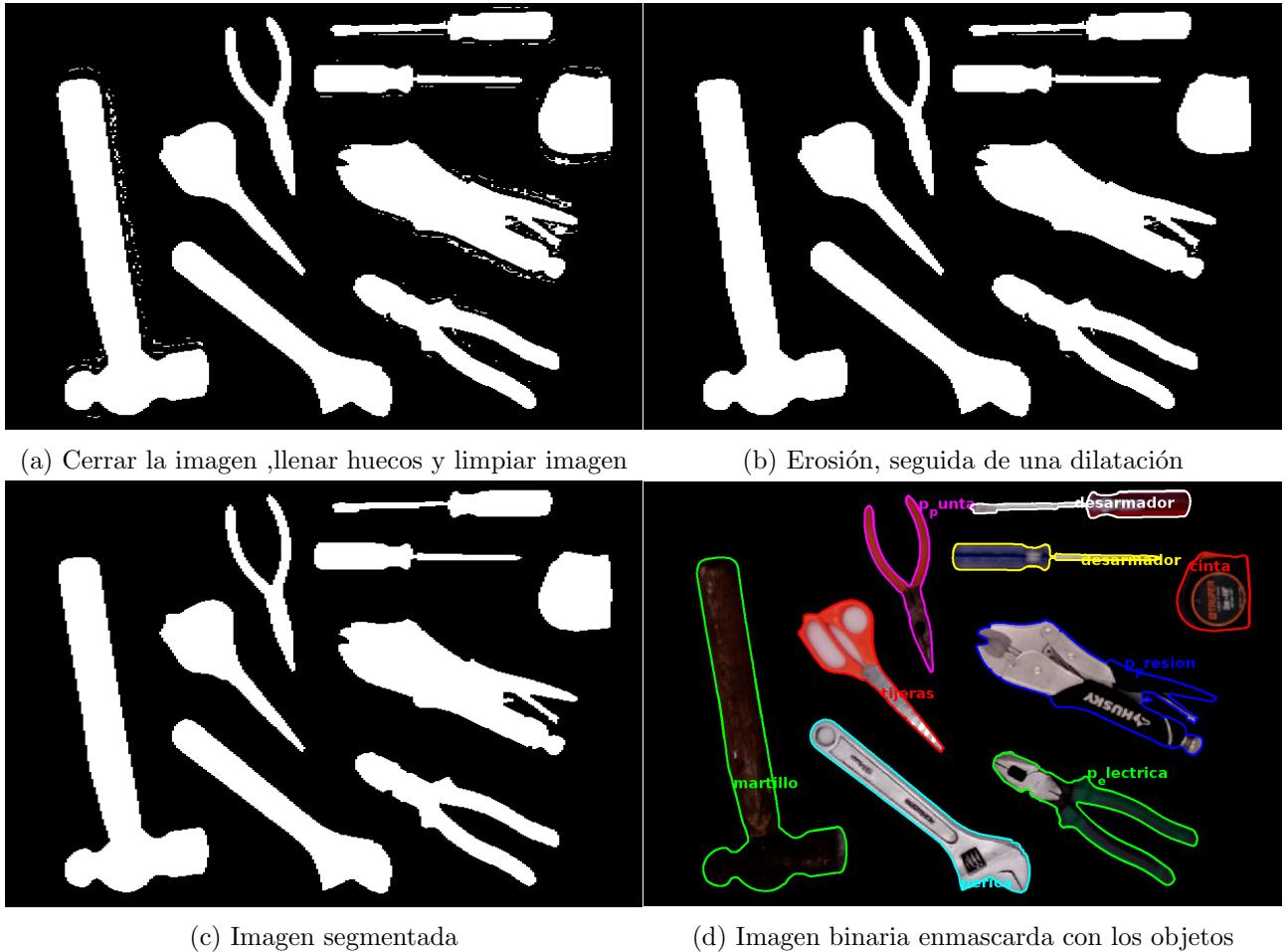


Figura 15: Proceso de segmentado última parte.

## 8. Conclusiones

Uno de los desafíos dentro del proyecto, fué el cómo lidiar con las partes cromadas dentro de un fondo claro como lo es el color blanco, usando técnicas de segmentación el objeto se perdía completamente en su forma particular que resultaría difícil el reconocimiento del objeto.

Para salir del problema se optó en cambiar la estrategia al usar la técnica de detección de bordes Canny Edges, la cuál nos permitió rescatar un poco la silueta de la imagen de interés.

Una de las ideas de seguir esta estrategia es hacer que el reconocimiento sea invariable en el fondo de la imagen dentro del ambiente controlado. Se probarón los fondos con los siguientes colores azul, verde, melón, blanco y claro.

Variando un poco el desempeño entre fondo y fondo. Esto debido a que la obtención de rasgos se realizó solo con el fondo blanco, es probable que al añadir más muestras con los diferentes fondos pueda cambiar el desempeño en la tarea de reconocimiento.

Estudios tempranos de Inteligencia Artificial buscaban duplicar los pensamientos humanos, pero ahora los estudios recientes muestran la tendencia en replicar el resultado y las computadoras actúan como sistemas expertos.



(a) Imagen objetos fondo negro



(b) Resultado fondo negro, hue 100 %



(c) Imagen objetos con fondo rojo



(d) Resultado fondo rojo, cerco convexo 89 %

Figura 16: Imágenes diferentes fondos.

Las computadoras son dispositivos simbólicos capaces de manipular cualquier tipo de símbolos siendo los números una clase importante, pero las computadoras son más generales que eso. Sabemos de la generalidad de la computación desde los tiempos de Alan Touring en los 1930's y se tienen intuiciones de que Babbage tuvo también estudios que fueron afirmados por Ada Lovelace, en 1842 Ada Lovelace escribió sobre la ingeniería analítica de Babbage que buscaba unir el mundo mecanico con el mundo de las cosas abstractas, en la psicología moderna es llamado The physical symbol system hypothesis y son las bases en que la Inteligencia Artificial funciona.

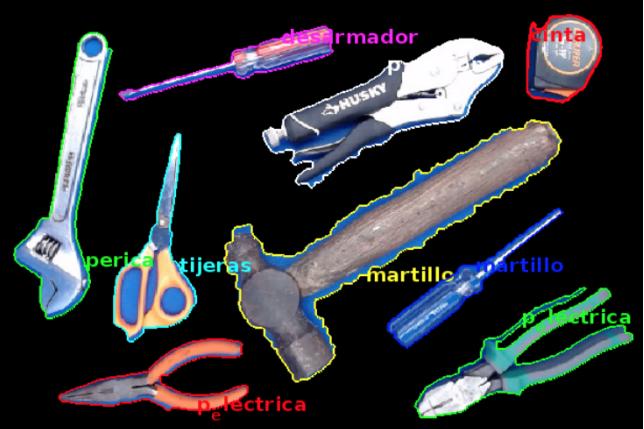
La Inteligencia Artificial como ciencia emplea el uso de computadoras para procesar conocimientos simbolicos usando métodos de inferencias logicas, en otras palabras, nos referimos a inferencia y no calculos que se piensa en el sentido tradicional, hablamos de conocimiento y no numeros como en la forma tradicional. [3]

Al poner conocimiento en programas computacionales que para los humanos nos es fácil o a veces un reto intelectual y el conocimiento que le pasamos es representativo en cierta forma particular.

Por otra parte, los sistemas expertos basados en conocimientos son aplicables en cualquier área en que sea especializado el conocimiento y sea rutinario la toma de decisiones, estrategias para resolver problemas, diagnosticos, .. etc. Siendo de gran ayuda en un gran rango de aplicaciones, el conocimiento



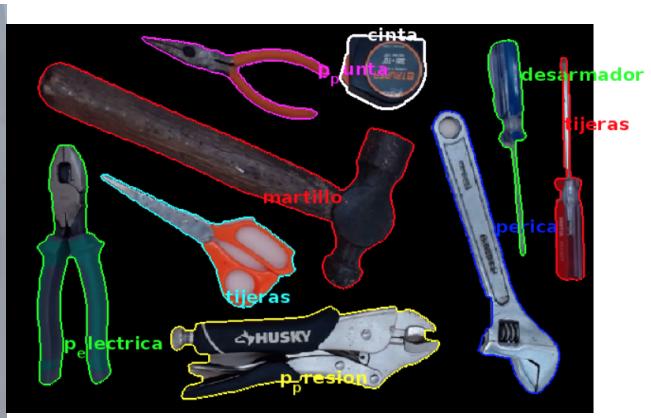
(a) Imagen objetos fondo azul



(b) Resultado fondo azul, rasgos geometricos 97 %



(c) Imagen objetos fondo claro



(d) Resultado fondo claro, cerco convexo 89 %

Figura 17: Proceso de segmentado última parte.

médico, en clasificación de un experto, puede estar reflejado con el funcionamiento de una red neuronal.

Para algún diagnóstico médico ó segmentadores más fáciles, pero igual de complicados como la segmentación en imágenes para la creación de sistemas autónomos.

Programas que se puedan ejecutar en cualquier computadora o dispositivo que permita la ejecución de código multiplataforma como python ó C++, que tienen capacidades altas de manipulación simbólica, la toma de decisiones es primordial en una inteligencia basada en conocimiento y esa manipulación simbólica es necesaria.

## Referencias

- [1] R. Aggarwal and A. Singhal. Augmented reality and its effect on our life. In *2019 9th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, pages 510–515, 2019.
- [2] CDW. logitech c920s HD. <https://www.cdw.com/product/logitech-hd-pro-webcam-c920s-webcam/5479466>, 2023. [Online; accessed 27-August-2023].
- [3] T. C. Chronicles. Artificial Inteligence. [https://www.youtube.com/watch?v=\\_S3mOV\\_ZF\\_Q](https://www.youtube.com/watch?v=_S3mOV_ZF_Q), 1984. [Online; accessed 27-August-2023].
- [4] M. N. R. Devi, A. Kumar, G. Swetha, U. S. Chavan, and V. M. Davasam. Cancer detection using image processing and machine learning. In *2022 International Conference on Artificial Intelligence and Data Engineering (AIDE)*, pages 96–100, 2022.
- [5] MatLab. edge. <https://la.mathworks.com/help/images/ref/edge.html#buo5g3w-1-threshold>, 2023. [Online; accessed 27-August-2023].
- [6] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield. Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness, 2017.
- [7] Tejas1415. Hu Invariant Moments in MATLAB. [https://raw.githubusercontent.com/Tejas1415/Hu-s-Invariant-Moments-in-MATLAB/master/hu\\_moments.m](https://raw.githubusercontent.com/Tejas1415/Hu-s-Invariant-Moments-in-MATLAB/master/hu_moments.m), 2023. [Online; accessed 29-Julio-2023].